09/422,018

AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	SIEVERT	Examiner:	Wood, W.
Serial No.:	09/422,018	Group Art Unit:	2193
Filed:	October 21, 1999	Docket No.:	RA-5236 (USYS.007PA)
Title:	A METHOD FOR IMPLEMENTING COMPONENT OBJECT MODEL INTERFACES		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on November 3, 2005.

By: Kathleen Klinkhammer
Kathleen Klinkhammer

TRANSMITTAL

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

We are transmitting herewith the attached:

☒ Transmittal sheet containing Certificate under 37 CFR 1.8.

☒ Appeal Brief pages 1-15.

☒ One Return Postcard.

☒ Please charge Deposit Account No. 50-0996 (USYS.007PA) in the amount of $500.00 for the filing of the brief in support of an appeal as indicated in 37 C.F.R. §1.17(c).

☒ Please charge/credit Deposit Account No. 50-0996 (USYS.007PA) for any deficiencies/overpayments.

Respectfully submitted,

CRAWFORD MAUNU PLLC
1270 Northland Drive, Suite 390
Saint Paul, MN 55120
651.686.6633

Name: LeRoy D. Maunu
Reg. No.: 35,274

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	SIEVERT	Examiner:	Wood, W.
Serial No.:	09/422,018	Group Art Unit:	2193
Filed:	October 21, 1999	Docket No.:	RA-5236 (USYS.007PA)

Title: METHOD FOR IMPLEMENTING COMPONENT OBJECT MODEL INTERFACES

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on November 3, 2005.

By: Kathleen Klinkhammer
Kathleen Klinkhammer

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an Appeal Brief submitted pursuant to 37 C.F.R. § 41.37 for the above-referenced patent application. Please charge Deposit Account No. 50-0996 (USYS.007PA) in the amount of $500 for this brief in support of appeal as indicated in 37 C.F.R. § 41.20(b)(2). If necessary, authority is given to charge/credit deposit account 50-0996 (USYS.007PA) any additional fees/overages in support of this filing.

I. Real Party in Interest

The real party in interest is Unisys Corporation having a place of business at Township Line and Union Meeting Roads, Blue Bell, PA 19424.

II. Related Appeals and Interferences

Appellant is unaware of any related appeals, interferences or judicial proceedings.

III. Status of Claims

Claims 1-17 are rejected and are presented for appeal. The appealed claims are in the attached Appendix of Appealed Claims.

IV. Status of Amendments

No amendments have been made to the claims after final rejection.

V. Summary of Invention

In one embodiment, the invention provides a computer-implemented method for implementing a hierarchy of component object model interfaces. The method includes defining a hierarchy of component object model interfaces (e.g., FIG. 2A, 202, 204, 206; p. 3, l. 1-14), wherein an interface at a lowest level (e.g., FIG. 2A, 206) of the hierarchy inherits from an interface at the highest level of the hierarchy (e.g., FIG. 2A, 202). A first template class is defined (e.g., FIG. 2B, 254; p. 11, l. 1-22) that is associated with the highest level of the hierarchy. A second template class is defined (e.g., FIG. 2B, 258; p. 11, l. 1-22) that inherits from the first template class and that is associated with the lowest level of the hierarchy. The second template class is instantiated with an interface as a template parameter (p. 11, l. 25 – p. 12, l. 19).

In another embodiment, the method includes defining a hierarchy of component object model interfaces (e.g., FIG. 2A, 202, 204, 206; p. 3, l. 1-14), wherein an interface at a lowest level (e.g., FIG. 2A, 206) of the hierarchy inherits from an interface at the highest level of the hierarchy (e.g., FIG. 2A, 202). A first template class is defined (e.g., FIG. 2B, 254; p. 11, l. 1-22) that is associated with the highest level of the hierarchy. A second template class is defined (e.g., FIG. 2B, 258; p. 11, l. 1-22) that inherits from the first template class and that is associated with the lowest level of the hierarchy. An interface of the lowest level of the hierarchy (e.g., FIG. 2A, 206, p.) is provided as a template parameter (e.g., p. 12, l. 10) to a template class (e.g., FIG. 2B, 256) directly inherited (p. 12, l. 12-15) by the second class (258).

Yet another embodiment provides a method that includes defining a hierarchy of component object model interfaces (e.g., FIG. 2A, 202, 204, 206; p. 3, l. 1-14), wherein an interface at a lowest level (e.g., FIG. 2A, 206) of the hierarchy inherits from an interface at the highest level of the hierarchy (e.g., FIG. 2A, 202). A first template class is defined (e.g., FIG. 2B, 254; p. 11, l. 1-22) that is associated with the highest level of the hierarchy. A second template class is defined (e.g., FIG. 2B, 258; p. 11, l. 1-22) that inherits from the first template class and that is associated with the lowest level of the hierarchy. The second template class is instantiated with a selected one (e.g., FIG. 2B, 258; p. 12, l. 1-19) of the component object model interfaces (e.g., FIG. 2B, 254, 256, 258) as a template parameter.

VI. Grounds of Rejection

Claims 1-17 stand rejected under 35 U.S.C. §103(a) as being unpatentable over "Gibbons" (US patent number 6,412,019 to Gibbons et al.) in view of "Shepherd" (Web pages entitled, "The Visual Programmer," by Shepherd).

VII. Argument

The rejection of claims 1-17 should be reversed because the Examiner has not established a *prima facie* case of obviousness of the claims under 35 U.S.C. §103(a) over the Gibbons-Shepherd combination.

The Examiner has failed to establish a *prima facie* case of obviousness of claims 1-17 over the Gibbons-Shepherd combination because all the limitations have not been shown to be suggested by the combination, a proper motivation for modifying Gibbons with teachings of Shepherd has not been provided, and no showing is made that any particular modification of Gibbons with teachings from Shepherd could be made with a reasonable likelihood of success.

Claims 1, 8, 16, and 17

The Examiner has failed to show that that the Gibbons-Shepherd combination suggests all the claim limitations. Various isolated keywords are extracted from the

prior art and used in an attempt to assemble elements into the claimed invention. However, even with the assembled elements from the prior art, the Examiner has assumed, but not shown, that some explicit limitations of the claims are present in the Gibbons-Shepherd combination. The general concepts taught by Gibbons and Shepherd are not shown to suggest the specific claim limitations.

Claim 1 includes limitations of defining a hierarchy of component object model interfaces, defining a first template class associated with the highest level in the hierarchy and a second template class that inherits from the first template class and is associated with the lowest level of the hierarchy. This combination limitations has not been shown to be taught or suggested by the Gibbons-Shepherd combination.

The cited portions of Gibbons appear to suggest interface inheritance (col. 3, ll. 24-30 and col. 6, ll. 25-35), and the cited portions of Shepherd appear to suggest COM classes, template inheritance, and template parameters that may include an interface (page 1, page 2, and page 9). However, there is no indication that a hierarchy of component object model interfaces have associated first and second template classes at the highest and lowest levels as claimed. Specifically, the cited section of Shepherd on page 9 illustrates an apartment class with a method that takes IDispatchImpl as a template parameter. There is no apparent suggestion of any second template class inheriting from the first template class and being associated with the lowest level of the hierarchy.

In addressing the limitations of defining the second template class (that inherits from the first template class and is associated with the lowest level of the hierarchy), the Examiner indicated, "Shepherd illustrated a first template class as discussed; additional template classes are based upon the same citing as software is implemented using any number of classes and thus template classes." This reason appears to say that additional template classes could be defined to meet the claim limitations of the second template class. However, this reasoning improperly relies on what could be done rather than what the cited teachings actually suggest. As stated above, there is no suggestion of the combination of limitations including defining a first template class associated with the highest level in the hierarchy and a second template class that inherits from the first template class and that is

associated with the lowest level of the hierarchy. Thus, the limitations of claim 1 are clearly not shown to be suggested by the Gibbons-Shepherd combination.

The Examiner has further failed to show that the Gibbons-Shepherd combination suggests the instantiation of the second template class with an interface as a template parameter. For purposes of illustration, pages 11 and 12 of the present specification provide a specific example of a hierarchy of component object model interfaces, associated template classes, and instantiation of a template class with an interface as a template parameter. No similarities appear to be suggested by the cited references. Shepherd is cited as teaching passing a template parameter. However, there is no apparent passing of an interface as a template parameter as is specifically claimed.

Claim 8 depends from claim 1, and the limitations are not shown to be suggested by the Gibbons-Shepherd combination for at least the reasons set forth above.

Claims 16 and 17 include similar limitations that are not shown to be suggested by the Gibbons-Shepherd combination.

The Office Action further fails to provide a proper motivation for combining Shepherd with Gibbons. The alleged motivation states that "it would have been obvious ... to implement Gibbons' hierarchy of interfaces with a COM base and a templating ability as found in Shepherd's teaching, thus developing an inheritance interface system, which is parameterized with templates ... because one of ordinary skill in the art would be motivated to provide a system utilizing COM as it is well known (and especially useful in the Microsoft world) and utilizing templates as they produce code which is very extensible while at the same time reducing bloated code." This alleged motivation merely alleges that COM was well known and that templating code makes extensible code. No evidence from the prior art has been provided that suggests making the specific combination of limitations as discussed in detail above. The Examiner must identify specifically clear and particular reasons, and provide evidence from the prior art, why one of ordinary skill in the art would have been motivated to select the references and combine (or modify) them (see, e.g., In re Dembiczak, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). In this

instance, the Examiner merely alleges general characteristics of COM and templates, without providing supporting reasons from the cited art. The alleged motivation is conclusory and therefore, improper and should be withdrawn.

The impropriety of the motivation for making the Gibbons-Shepherd is further demonstrated by Gibbons' and Shepherd's teachings. Gibbons teaches interface inheritance in a C++ implementation (Abstract; col. 4, l. 53 – col. 6, l. 67), and Shepherd teaches that "critical chunks of code necessary for COM classes to work are brought in via template inheritance rather than normal C++ implementation inheritance" (page 2, last paragraph). Thus, Shepherd apparently teaches away from a modification of Gibbons' approach. Furthermore, the Examiner has not provided any indication of which specific elements of Gibbons' might be modified with specific elements of Shepherd to make the claimed invention and to overcome the apparent inapplicability of Shepherd to Gibbons. Therefore, the alleged motivation is improper.

The rejection of claims 1, 8, 16, and 17 should be reversed because the Examiner failed to establish a *prima facie* case of obviousness.

Claims 2 and 3

Claims 2 and 3 depend from claim 1 and further refine the limitations discussed above. Therefore, the Examiner failed to establish a *prima facie* case of obviousness for claims 2-3 for at least the reasons set forth above. In addition, the rejections of these claims allege that certain features are "inherent." As explained below, the rejections based on inherency are improper.

The Examiner has fails to provide evidence that the Gibbons-Shepherd combination necessarily possesses the claim limitations. Thus, the alleged inherency is improper.

Claims 2 and 3 set forth mutually exclusive limitations, and it would present a contradiction for both sets of limitations to necessarily be present in the Gibbons-Shepherd combination. In claim 2, the second template class inherits directly from the first template class, and in claim 3, the second template class inherits indirectly from the first template class. Those skilled in the art would recognize that a second

template class could not both inherit directly and inherit indirectly from a first template class. Yet, in alleging inherency the Examiner is asserting that the Gibbons-Shepherd combination necessarily has a second template class that both directly and indirectly inherits from a first template class. Since the limitations of claims 2 and 3 are mutually exclusive, the allegation of inherency for claims 2 and 3 in the Gibbons-Shepherd combination is clearly improper. Even considered individually, the Examiner has failed to present evidence that either of the claimed implementations is necessarily present in the Gibbons-Shepherd combination.

The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. *In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993). "To establish inherency, the extrinsic evidence must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient." *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999) (citations omitted). "In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art." *Ex parte Levy,* 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990) (emphasis in original) (MPEP 2112). The inherency rejection is improper because no technical reasoning has been provided to support the allegation. Furthermore, the subject limitations are not believed to be necessary in the teachings of either Gibbons or Shepherd. Therefore, the rejections based on inherency are improper.

The rejection of claims 2-3 should be reversed because the Examiner failed to establish a *prima facie* case of obviousness.

Claims 4, 5, 6, and 7

Claims 4-7 depend directly or indirectly from claim 1 and further refine the limitations discussed above. Therefore, the Examiner failed to establish a *prima*

facie case of obviousness for claims 4-7 for at least the reasons set forth above. In addition, the rejections of these claims allege that certain features are "inherent." The Examiner has failed to show that the claim limitations are inherent in the Gibbons-Shepherd combination.

Claims 4-7 include various limitations related to intermediate classes between the first and second template classes. The Examiner has failed to show that the Gibbons-Shepherd combination necessarily includes the claimed intermediate classes (and hasn't shown the presence of a second template class as explained above). From the prior art of record it appears possible that the Gibbons-Shepherd combination has only a first template class and does not necessarily include intermediate template classes as claimed. Therefore, the rejections fail to establish inherency of the claim limitations.

The rejection of claims 4-7 should be reversed because the Examiner failed to establish a *prima facie* case of obviousness.

Claim 9

Claim 9 depends from claim 1 and includes numerous additional limitations. The Examiner has not cited any additional teachings of the Gibbons-Shepherd combination as suggesting these limitations. Therefore, for the reasons set forth above for claim 1 and because the Gibbons-Shepherd combination does not appear to suggest the limitations of claim 9, the Examiner failed to establish a *prima facie* case of obviousness. The rejection of claim 9 should be reversed.

Claim 10

Claim 10 includes limitations that relate to defining ActiveX Template Library interface maps in the first template class and in the second template class. The Examiner has failed to show where Shepherd suggests interface maps in both the first and second template classes as claimed. The general observation that ActiveX Template Library supports interface maps does not by itself suggest the specifically claimed manner in which the interface maps are used. That is, the first and second template classes are associated with the highest and lowest levels of the hierarchy,

and both the first and second template classes have interface maps defined therein. The only evidence provided by the Examiner is the assertion that ATL includes interface maps. Therefore, the Examiner has failed to show that the Gibbons-Shepherd combination suggests the limitations of claim 10.

Claims 11, 12, 13, 14, and 15

Claims 11-15 depend directly or indirectly from claim 10 and include limitations similar to those of claims 4, 5, 8, 9, and 10 as discussed above. Therefore, for the reasons set forth above, the Examiner failed to establish a *prima facie* case of obviousness for claims 11-15, and the rejection should be reversed.

VIII. Conclusion

In view of the above, Appellant submits that the rejections are improper, the claimed invention is patentable, and that the rejections of claims 1-17 should be reversed. Appellant respectfully requests reversal of the rejections as applied to the appealed claims and allowance of the entire application.

Respectfully submitted,

CRAWFORD MAUNU PLLC
1270 Northland Drive, Suite 390
Saint Paul, MN 55120
(651) 686-6633

By: _____
Name: LeRoy D. Maunu
Reg. No.: 35,274

APPENDIX OF APPEALED CLAIMS FOR APPLICATION NO. 09/422,018

1. (original) A computer-implemented method for implementing a hierarchy of component object model interfaces, comprising:

defining a hierarchy of component object model interfaces, wherein an interface at a lowest level of the hierarchy inherits from an interface at the highest level of the hierarchy;

defining a first template class that is associated with the highest level of the hierarchy;

defining a second template class that inherits from the first template class and is associated with the lowest level of the hierarchy; and

instantiating the second template class with an interface as a template parameter.

2. (original) The method of claim 1, wherein the second template class inherits directly from the first template class.

3. (original) The method of claim 1, wherein the second template class inherits indirectly from the first template class.

4. (original) The method of claim 1, further comprising defining a plurality of intermediate classes in a single inheritance arrangement, one of the intermediate classes inheriting from the first template class, and the second template class inheriting from another one of the intermediate classes.

5. (original) The method of claim 4, wherein one or more of the intermediate classes are template classes.

6. (original) The method of claim 1, further comprising defining an intermediate class, the intermediate class inheriting from the first template class, and the second template class inheriting from the intermediate class.

7. (original) The method of claim 6, wherein the intermediate class is a template class.

8. (original) The method of claim 1, wherein the interface provided as the template parameter is an interface at the lowest level of the hierarchy.

9. (original) The method of claim 1, further comprising:

extending the hierarchy of component object model interfaces to include a new interface defined at the lowest level of the hierarchy, wherein the new interface inherits from the interface at the highest level of the hierarchy;

defining a third template class that inherits from the first template class and is associated with the new interface defined at the lowest level of the hierarchy; and

instantiating the third template class with the new interface as a template parameter.

10. (original) The method of claim 1, further comprising defining ActiveX Template Library interface maps in the first template class and in the second template class, respectively.

11. (original) The method of claim 10, further comprising defining a plurality of intermediate classes in a single inheritance arrangement, one of the intermediate classes inheriting from the first template class, and the second template class inheriting from another one of the intermediate classes.

12. (original) The method of claim 11, wherein one or more of the intermediate classes are template classes.

13. (original) The method of claim 12, further comprising defining ActiveX Template Library interface maps in the respective intermediate template classes.

14. (original) The method of claim 13, wherein the interface provided as the template parameter is an interface at the lowest level of the hierarchy.

15. (original) The method of claim 14, further comprising:

extending the hierarchy of component object model interfaces to include a new interface defined at the lowest level of the hierarchy, wherein the new interface inherits from the interface at the highest level of the hierarchy;

defining a third template class that inherits from the first template class and is associated with the new interface defined at the lowest level of the hierarchy; and

instantiating the third template class with the new interface as a template parameter.

16. (original) A computer-implemented method for implementing a hierarchy of component object model interfaces, comprising:

defining a hierarchy of component object model interfaces, wherein an interface at a lowest level of the hierarchy inherits from an interface at the highest level of the hierarchy;

defining a first template class that is associated with the highest level of the hierarchy;

defining a second class that inherits from the first template class and is associated with the lowest level of the hierarchy; and

providing an interface of the lowest level of the hierarchy as a template parameter to a template class directly inherited by the second class.

17. (previously presented) A computer-implemented method for implementing a hierarchy of component object model interfaces, comprising:

defining a hierarchy of component object model interfaces, wherein an interface at a lowest level of the hierarchy inherits from an interface at the highest level of the hierarchy;

defining a first template class that is associated with the highest level of the hierarchy;

defining a second template class that inherits from the first template class and is associated with the lowest level of the hierarchy; and

instantiating the second template class with a selected one of the component object model interfaces as a template parameter.

APPENDIX OF EVIDENCE FOR APPLICATION NO. 09/422,018

Appellant is unaware of any evidence submitted in this application pursuant to 37 C.F.R. §§ 1.130, 1.131, and 1.132.

APPENDIX OF RELATED PROCEEDINGS FOR APPLICATION NO. 09/422,018

Appellant is unaware of any related appeals, interferences or judicial proceedings.